UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/449,021 | 11/24/1999 | HELMUT EMMELMANN | EMMEL-1000US0 | 5718 |

28554 7590 07/01/2009

Vierra Magen Marcus & DeNiro LLP
575 Market Street, Suite 2500
San Francisco, CA 94105

| EXAMINER | |
|---|---|
| KENDALL, CHUCK O | |
| ART UNIT | PAPER NUMBER |
| 2192 | |
| MAIL DATE | DELIVERY MODE |
| 07/01/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| | | | |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.** | **Applicant(s)** | |
| | 09/449,021 | EMMELMANN, HELMUT | |
| | **Examiner** | **Art Unit** | |
| | CHUCK O. KENDALL | 2192 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on _06 January 2009_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) _1 – 6, 8, 22 – 33, 41 – 43, 51 – 96, and 114 – 128_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1 – 6, 8, 22 – 33, 41 – 43, 51 – 96, and 114 – 128_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _24 November 1999_ is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _10/17/08_.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

## DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 12/30/08 has been entered.

2. Claims 1 – 6, 8, 22 – 33, 41 – 43, 51 – 96, and 114 – 128 are pending in this application.

### *Claim Rejections - 35 USC § 101*

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 6, 8, 114-124 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims draws limitations to, " a system for " use in a data network etc. The system as claimed doesn't appear to include any associated

hardware and hence falls under software per se type subject matter. Software per se type subject matter isn't considered statutory subject matter as it requires an underlying hardware to be able to carry out its intended functionality.

## ***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1 – 6, 8, 22 – 33, 41 – 43, 51 – 96, and 114 – 128 are rejected under 35 U.S.C. 103(a) as being unpatentable over Web Writer: A Browsers -Based Editor for Constructing Web Applications Arturo et al. Published May 1996 (Hereinafter "Web Writer") in view of Web Writer II, Arturo et al. Published 1997.

Regarding claims 1, 6, 22, 26, 51, 59, 74, 90, 114, 125, a computer-readable medium encoded with computer programs having executable instructions software development system for editing software applications that run on a data network and a client computer, wherein the client computer runs a browser program, and whereupon request by the browser program at least one of the applications generates generated documents for display by the browser program on a display device and responds to the request with the generated documents page comprising:

a page document generator program running at least part of one of the applications being edited and generating the generated documents said generated documents including additional editing features for interpretation by the browser program (page 9, column 1, see Web Writer page Generator); and

an editor program dynamically operating on the generated documents pages displayed by the browser program via the editing features (page 9, column 1, see Web Writer page Generator, see on the fly, also see page 2, first paragraph for buttons and computing content on the fly for components as claimed in claim 59).

Although, Web Writer doesn't disclose having a server, Web WriterII discloses Web Writer coupling a WebServer in its implementation (section 1.2). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made combine Web Writer and Web Writer II, because it would enable making changes on the fly remotely.

Regarding claim 2, a computer-readable medium software development system as claim 1, further encoded with further comprising a plurality of components, and wherein developed the software applications comprise at least one page document template capable of containing components, and wherein the editor provides features to insert, modify and delete a component on at least one page document template, and wherein the page-document generator executes selected components on page document templates (page 2, Column 1, see Heading The Web Writer Application Model, shows template page).

Regarding claim 3, a computer-readable medium software development system as in claim 2, wherein at least one of the components reacts interactively on user input by executing instructions of said component on the server computer (see, FIG. 2).

Regarding claim 4, a computer-readable medium software development system as in claim 3, wherein at least one of the components contains at least one other component (FIG. 2, also see FIG. 4).

Regarding claim 5, a computer-readable medium software development system as in claim 3, wherein the set of components on pages documents generated from at least one page document template can vary for different requests of said page document template (page 2, Column 1, see Heading The Web Writer Application Model, shows template page, also see page 9).

Regarding claim 8, the development system of claim 6, wherein a component is nested within a component (Web Writer, see FIG. 4).

Regarding claim 23, a computer readable medium as in claim 22, wherein the editor program operates functional applications in an edit mode permitting editing directly in the web browser (page 2, Column 1, see Heading The Web Writer Application Model, shows template page).

Regarding claim 24, a computer readable medium as in claim 23 wherein at least one of the components contains instructions and can react on subsequent document requests containing user responses by executing selected instructions of said component (See Web Writer page 2 – 9 and et. seq).

Regarding claim 25, a computer readable medium as in claim 24, of component classes, each component class implementing one component kind (Web Writer page 2 – 9, et seq); and

a parser program able to detect components marked on document templates (Web Writer page 2 – 9, et seq);

wherein the document generator program works upon a document request using component classes to generate browser code (Web Writer page 2 – 9, et seq); and

wherein the editor program is capable of showing a menu of components for insertion into the document templates (See Web Writer page 2 – 9 see editor, equivalent function).

Regarding claim 27, the system of claim 26, wherein the first document includes at least one component being executed by the first software program (See Web Writer page 2 – 9 and et. seq).

Regarding claim 28, the system of claim 27, wherein the second document includes handles and choosing one of the handles selects a component for an editing operation (Web Writer page 4, see clicking on its handles).

Regarding claim 29, the system of claim 28, wherein at least one handle indicates the position of at least one component contained in the first document and said editing operation includes modifying the component, deleting the component, and displaying information regarding the component (Web Writer page 4, see clicking on its handles).

Regarding claim 30, the system of claim 26, wherein the features include scripts (See Web Writer page 2 – 9 and et. seq).

Regarding claim 31, the system of claim 30, wherein the scripts encapsulate information from the first document (See Web Writer page 2 – 9 and et. seq)

Regarding claim 32, the system as in claim 26, wherein the features incorporate information regarding the first document into the second document (See Web Writer page 2 – 9 and et. seq).

33. (currently amended) A system as in claim 32, wherein the information incorporated into the second document is used on the client computer in order to send change requests for the first document to the server computer (Web WriterII, see page 1510

section 4.1).

Regarding claim 41, (currently amended) a computer-readable medium as in claim 1, the editor program comprising a client part for execution on the client computer (Web WriterII, see page 1510 section 4.1).

Regarding claim 42, a computer-readable medium as in claim 41, wherein the client part comprises instructions for execution during editing that are automatically downloaded from the server computer in a request prior to editing (Web WriterII, see page 1510 section 4.1).

Regarding claim 43, a system computer-readable medium as in claim 26, additionally comprising at least one script for automatic download to the client that works in cooperation with the second document to permit editing of the first document (Web WriterII, see page 1510 section 4).

52. (previously amended) The system of claim 51, wherein at least some

components include fourth program instructions including steps to generate browser code for transmission to the first software program in response to a request from the first software program, wherein the browser code can differ for multiple requests for the same document template (See Web Writer page 2 – 9 and et. seq).

53. (currently amended) A system as in claim 52 having a data network, coupling the computer and a client computer, the first program running on the client computer (Web WriterII, see page 1508 section 1.2).

54. (currently amended) A system as in claim 52 wherein second documents include HTML pages with embedded scripts(See Web Writer page 2 – 9 and et. seq).

55. (currently amended) The system of claim 52, wherein the editing functions includes adding a component to a document template, removing a component from a document template, and modifying a component on a document template (See Web Writer page 2 – 9 , see editor).

56. (previously added) The system of claim 52, further comprising a fifth software program used by the second software program while processing selected document requests, the fifth software program

including fifth instructions for generating generated documents from document templates thereby calling fourth program instructions (See Web Writer page 2 – 9 , see templates and generator).

57. (previously added) The system of claim 56, wherein the generated document includes, if requested in edit mode, edit features for interpretation by the first software program (See Web Writer page 2 – 9 , see editor).

58. (previously amended) The system of claim 56 further comprising instructions to allow the user to click on the generated document to select items to perform edit functions on (See Web Writer page 2 – 9 , see editor).

60. (currently amended) The software development system of Claim 59 comprising a data network which couples a server computer and a client computer, the document generator program running on the server computer, the editor program at least partly running on the client computer(Web WriterII, see page 1510 section 4).

61. (currently amended) The software development system of claim 60 comprising fourth instructions for execution during document

generation to collect edit- information for use by the editor program (See Web Writer page 2 – 9 , see editor).

62. (currently amended) The software development system of claim 60, wherein the editor program uses a web browser for displaying said view(See Web Writer page 2 – 9, shows web browser).

63. (currently amended) The software development system of claim 60, comprising instructions for automatically repeating the requesting that the document generator processes the dynamic web document when if required (See Web Writer page 2 – 9, shows web browser).

64. (currently amended) The software development system of Claim 59 further comprising a plurality of components including at least one component marked on said dynamic web document and including instructions for use by the document generator program to generate browser code (See Web Writer page 2 – 9, see on the fly).

65. (currently amended) The software development system of claim 64, wherein the editor program uses a web browser for displaying said view (See Web Writer page 2 – 9, shows web browser).

66. (currently amended) The software development system of claim 64, wherein the modification functions includes insertion of a

component, detete deletion of a component, and ~~modify~~ modification of a component (See Web Writer page 2 – 9, shows see editor).

67. (previously added) The software development system of claim 59, wherein said view looks, except for editing features, similar to the end-user view of the generated document(See Web Writer page 2 – 9, shows see editor).

68. (currently amended) The software development system of claim 59 comprising sixth instructions to collect edit-information for use by the editor program, said sixth instructions for execution during document generation(See Web Writer page 2 – 9, shows document generator).

69. (currently amended) The software development system of claim 68, wherein the editor program uses the edit-information to correctly modify the dynamic web document (See Web Writer page 2 – 9, shows see editor).

70. (currently amended) The software development system of claim 69, further comprising a plurality of components wherein the edit-information comprises position information on selected components marked on the dynamic web document (See Web Writer page 2 – 9, shows dynamic web content).

71. (currently amended) The software development system of claim

59, wherein the editor program uses a web browser for displaying said view(See Web Writer page 2 – 9, shows see editor).

72. (currently amended) The software development system of claim 71, wherein the first instructions comprise seventh instructions for initiating a reload in the browser (See Web Writer page 2 – 9, shows browser).

73. (currently amended) The software development system of claim 59 wherein the editor program comprises eighth instructions to display information on at least one element of at least one dynamic web document, that is replaced during document generation, without requesting that the document generator program generates a document (See Web Writer page 2 – 9, shows generator).

75. (currently amended) The system as in claim 74, wherein the editing functions comprises adding a component, modifying a component, and deleting a component (See Web Writer page 2 – 9, shows see editor).

76. (currently amended) The ~~software development~~ system as in claim 74, wherein tag syntax is used to denote at least one component on at

least one document template, whereby the tag name identifies the component kind(See Web Writer page 2 – 9, shows use of templates).

77. (currently amended) The system of claim 74 comprising a server computer coupled to a client computer by a data network, the document generator program running on the server computer, and the editor program running, at least partly, on the client computer (Web WriterII shows Web server and editor, also see Web Writer).

78. (currently amended) The ~~software development~~ system as in claim 74, wherein at least one component; that can react interactively on subsequent document requests can be excluded from said generated document upon selected document requests for said document template (See Web Writer page 2 – 9, shows generator and templates).

79. (currently amended) The ~~software development~~ system as in claim 78 comprising third instructions to prevent excluded components from reacting on subsequent document requests (See Web Writer page 2 – 9).

80. (currently amended) A ~~software development~~ system as in claim

79, said third instructions comprising fourth instructions to, upon a first document request, store information in session memory on some of the components that are present on the document generated based on the first document request, and fifth instructions to, upon subsequent document requests, only react on components that have been remembered in session memory thereby avoiding tampering with excluded components on the side of the first program (See Web Writer page 2 – 9).

81. (currently amended) A ~~software development~~ system as in claim 74 wherein at least one first component contains sixth instructions to decide upon a request for said document template about exclusion of components nested inside the first component from the generated document(See Web Writer page 2 – 9, shows Templates).

82. (currently amended) A ~~software development~~ system as in claim 74 the system ~~able to~~ provide providing an editable view taking the varying set of components into account (See Web Writer page 2 – 9, shows editor and display/interface).

83. (currently amended) A ~~software development~~ system as in claim 74 ~~system able to~~ provide providing an editable view that includes and

excludes selected components on different requests for said document template ~~similarly as~~ similar to the end user's view of the document template (See Web Writer page 2 – 9, shows editor and template).

84. (currently amended) A ~~software development~~ system as in claim 74 wherein a document generated for at least one document template contains more components than the document template for at least one document request (See Web Writer page 2 – 9, shows templates).

85. (currently amended) The ~~software development~~ system as in claim 74, wherein multiple instances of at least one third component denoted on the document template can be included in at least one of the documents generated from said document template(See Web Writer page 2 – 9, shows templates).

86. (currently amended) The system as in claim 74, comprising seventh instructions to assign a unique identifier to each component instance of at least one seventh component, whereby the seventh component includes eighth instructions to qualify names generated into the browser code with the unique identifier

(See under the heading, The Web Writer Application Model, " a set of named form elements used as variables").

87. (currently amended) A system as in claim 74, wherein at least one fourth component contains ninth instructions to decide how many instances of components nested inside the fourth component are included in the documents generated from said document template (See under the heading, The Web Writer Application Model, one or more dynamic areas on each template page," Examiner interprets this to be nested components).

88. (currently amended) A system as in claim 74 the editor program able to provide an editable view that includes multiple instances of selected components similar to the end user's view of the document template (See under the heading, Scripts and Formating for Output areas, multiple instances of variables are disclosed).

89. (currently amended) A system as in claim 74 wherein at least one sixth component includes tenth instructions to display the sixth component, the tenth instructions being used to generate browser code for displaying the sixth component during editing as well as during normal use of the component (See Web Writer page 2 – 9,

shows browser).

91. (currently amended) The system as in claim 90 wherein the editor program includes instructions to display at least two windows, a first browser window displaying said document and a second window for displaying information on an element contained in said document(See Web Writer page 2 – 9).

92. (currently amended) The system as in claim 90 comprising a second software program having instructions for storing modifications on said document in cooperation with the first software program (See Web Writer page 2 – 9, see editor).

93. (currently amended) The system ~ as in claim 92 further comprising a third program having instructions for transforming an original document into the document, the browser displaying the document as said view looking similar to the original document and interpreting editing features contained in the document (See Web Writer page 2 – 9, see editor).

94. (currently amended) The system ~ as in claim 93 wherein said original document is a dynamic document having components denoted thereon, the third software program comprising instructions for generating browser code in cooperation with selected instructions

contained in the components (See Web Writer page 2 – 9, shows browser).

95. (currently amended) The system ~ as in claim 94 comprising a client computer connected to a server computer via a data network, wherein the browser together with the first software program is running on athe client computer and the second and the third software program run on the server computer (See Web Writer page 2 – 9).

96. (currently amended) The system as in claim 90 wherein links contained in said document stay functional allowing the user to browse and edit at the same time (See Web Writer page 2 – 9, see editor).

115. (previously amended) The system of claim 128 wherein first features include fourth program instructions for passing information to the editor (See Web Writer page 2 – 9, see editor).

116. (currently amended) The system of claim 115 wherein at least part of said information is collected during execution of selected components on the server compute r(See Web Writer page 2 – 9).

117. (currently amended) The system of claim 115 wherein said information is transmitted from the server computer to the client computer (See Web Writer II et seq.).

118. (previously amended) The system of claim 115 wherein said information includes attribute values of said component (See Web Writer page 2 – 9, see editor).

119. (previously amended) The system of claim 128 wherein first features include fifth instructions that display additional editing features of the component during editing (See Web Writer page 2 – 9, see editor).

120. (previously added) The system of claim 119 wherein said editing features include handles (See Web Writer page 2 – 9, see editor).

121. (previously amended) The system of claim 128 wherein first features include an extension for use by the editor, said extension for enabling editing of an attribute value of the components(See Web Writer page 2 – 9, see editor).

122. (currently amended) The system of claim 121 wherein said extension enables display of a page for editing the-attribute values of the components (See Web Writer page 2 – 9, see editor).

123. (previously amended) The system of claim 128 wherein at least one component is denoted on at least one document template using tag syntax, whereby the tag name identifies a component kind (See Web Writer page 2 – 9,see templates).

124. (previously amended) The system of claim 114 containing at least one component wherein second program instructions are used to generate browser code for displaying the component during editing

and during normal use (See Web Writer page 2 – 9, see browser).

126. (currently amended) The method of claim 125 wherein the running step and the displaying step are repeated after the modification function (See Web Writer page 2 – 9, see editor).

127. (previously added) The method of claim 125 further comprising collecting edit information for use by the identifying step (See Web Writer page 2 – 9, see editor).

128. (previously added) The system of claim 114 additionally comprising a plurality of document templates with components denoted thereon, whereby the browser code generated by the components can vary for different requests of the same document template (See Web Writer page 2 – 9, see editor, generator and browser and templates as well).

### ***Response to Arguments***

Applicant's arguments filed 12/30/08 have been fully considered but they are not persuasive.

Argument (1), Applicant argues on page 2, of his response in the ("Overview" section) as well as in his response to the 103 rejection section on page 10, paragraph "B", that Web Writer only teaches static parts of each page created using a text editor and dynamic parts using placeholder for dynamic output which is replaced at runtime and only runs the edited content only after editing, versus running/executing the application during editing as argued by Applicant.

Response (1), in responding Applicant's plain language of claims as disclosed in claim 1, 6 and 22, doesn't preclude, include or exclude when execution or running of the editor as taught by Web Writer is performed. Running saved content such as a template on the fly, in one or more steps isn't disclosed as not meeting Applicant's claimed limitations as recited in Applicant's claims.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. not running only after editing and only running during editing) are not recited in the rejected claim(s), e.g. claim 1. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Examiner believes that Web Writer does in fact disclose this limitation in his disclosure and that Applicant's claim doesn't include or exclude any of the teachings as provided by Web Writer thereto.

In Web Writer in his introduction, Web Writer discloses construction of an interactive application which includes images, buttons and content that is "computed, on the Fly", by the Web Writer script, Emphasis added. In fact during editing since Web Writer does include buttons and components within the interactive template during editing, it is actually quite inherent that the button is an executable component and hence this is being performed during editing.

In the 2nd Paragraph Web Writer also discloses that the Web Writer Editor runs in the Browser and in addition in paragraph 4, Web

Writer discloses that the Page Generator, "creates new pages as an application runs", Emphasis added.

As disclosed above, contrary to Applicant's argument that Web Writer doesn't execute the editor while running, it shows above that in Web Writer the Generator creates pages as an application is run, as well as in page 4 of Web Writer, under the Heading, "Creating the Template pages", Web Writer is said to be an editor that runs in a Web browser, which seamlessly generates/creates web pages while the application is run.

Argument (2), Applicant argues on page 11 of his response that in claim 6, Web Writer merely argues that dynamic content as taught, shows the script name and a formatting string and so usually look very different from the appearance of a component generated page and that his claimed limitations requires that the components have a similar appearance.

Response (2), Applicant's plain language of claims 6 and 26, merely adds the limitation of having a similar appearance to the generated page with the addition of the editing features.

Examiner doesn't see this to be any different from the edited template in Web Writer which contains both the static and dynamic portions that are edited as well as the created pages which are run in the browser as taught by Web Writer on page 2, 2nd paragraph.

The extra limitation of claims 6 and 26 including the like appearance to the editor as claimed by applicant doesn't, show any distinct or distinguishing functionality as compared to Web Writer and doesn't exclude or preclude any likeness of Web Writers edited content to the content presented at runtime by the browser and hence claims are essentially the same.

Regarding Applicant's argument in claims 22, examiner has already addressed it as applicant simply rehashes the same arguments as presented in claim 1, that Web Writer doesn't execute part of the edited application during editing.

Regarding Applicant's argument in claim 51, Applicant asserts that claims 51 and 114 differ from claim 1 and shouldn't be rejected on under the same grounds, adding that the difference between claims is the recitation of the term "components".

Examiner still maintains claims are rejected under the same grounds as Web Writer, the prior art of record also recites components such as buttons and contents computed on the fly on page 2, first paragraph of his disclosure, and this is what is also cited as teaching Applicant's claimed limitations in claim 1.

Regarding Arguments in claim 59, Applicant essentially claims already addressed limitation as present above in Response (1).

Regarding Applicant's argument in claim 74, Applicant argues that, neither Web Writer I or Web Writer II discloses wherein the set of generated documents can very for different document requests and components having the ability to cooperate with the editor.

Examiner maintains that this is taught by Web Writer I. In Web Writer on page 2 of his disclosure, the editor cooperates with the browser as well as the page generator, seamlessly allowing the user to interactively construct application without the need to learn HTML as disclosed in page 2, first paragraph of Web Writer I.

Regarding claim 90, Applicant argues that the editor edits a document being actively displayed and includes a client part which initiates editing functions when the user clicks on the displayed content.

In Web Writer on page 3, under the heading, "...The Web Writer Application Model", Web Writer discloses that, "specifications of locations within the page, called *dynamic areas,* where computed content should be inserted at runtime to produce the final page that will be displayed to the user" Emphasis added. As disclosed the edited content i.e. *inserted at runtime* as disclosed above is displayed to the user, Examiner doesn't see how this is any different from Applicant's claimed limitation in claim 90.

Regarding Applicant's argument in claim 125, claim 125 essentially differs from claim 1, by it being a method claim rather than claim 1 being an apparatus claim and shouldn't be rejected using similar recitations from Web Writer.

Examiner disagrees, although the preamble differs by falling under a different category of statutory subject matter i.e. method and apparatus, both claims essentially disclose similar embodiments. Applicant's apparatus simply encodes instructions and steps, which are the same steps and instructions as disclosed in the method claim recited in claim 125, and which produce the same output as what is claimed in claim 125 and hence the grounds of rejections remains the same.

Regarding argument in claim 2, 3 - 5, for component see Web Writer I page 2, first paragraph for buttons and computer content on the fly, which is contained in the template as interpreted by Examiner.

Regarding claims 4-5, Applicant argues Web Writer doesn’t disclose wherein components contains at least one other component.

Web Writer discloses within the template are dynamic areas, see Web Writer page 2, Examiner interprets that to be components within components, as applicants claim doesn't preclude or exclude the definition of components as claimed.

Regarding claims 41 and 42, Applicant is rehashing previous arguments as previous addressed above in Response (1). Please see response to arguments above.

Regarding claim 23, rehashes arguments as previously addressed in response to arguments in claim 90, please see response above.

Regarding claim 24, Applicant argues that Web Writer’s dynamic areas are just used for processing single requests, and do not process multiple request as recited in claim 24.

Applicant’s plain language of claims discloses reacting to "subsequent document requests", and not multiple requests as argued by applicant.

In response to applicant's argument that the references fail to show certain features of applicant’s invention, it is noted that the features upon which applicant relies (i.e., multiple request) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In a system which utilizes user interaction to build generated pages using templates, it would be inherent for that document to be requested numerous times based on what template is being used.

Therefore Applicant argument is moot based on unclaimed merits of the limitations.

Regarding claim 25, Applicant is rehashing previous arguments as previously addressed above.

Regarding claims 27 - 33 and 43, Applicant is rehashing previous arguments as previously addressed above. Regarding handles, on page 20 of Applicant’s argument, Applicant agrees handles are being taught in Web Writer, however Applicant asserts that the handles as taught in Web Writer are only present during editing inside the a document rather than dynamically. Examiner again disagrees, Web Writer on page 2 of his disclosure, shows dynamic areas within the template which is edited and also executed.

Regarding claims 52 – 58, Applicant simply rehashes arguments as presented in claim 51. Please refer to argument above as previously addressed.

Regarding claims 60-69 Applicant again rehashes arguments previously presented and addressed in claims 1, regarding the editor and dynamic content. Applicant argues that the web generator as taught by Web Writer doesn't disclose generating documents from dynamic web documents. Web Writer I on page 2 discloses "*…Web Writer Page Generator, … creates new pages as an application runs…",* Therefore this clearly discloses applicants claimed limitations.

Also Applicant additionally argues in claim 66, that Web Writer doesn't disclose deleting a component.

Examiner disagrees, under the heading, Web Writer editor

Web Writer discloses, "... are three types of actions: changes to the document state, such as *insert, select, modify, cut, copy, and* paste; file operations such as save and load; and changes to the document view, such as hiding handles or showing handles…." Emphasis added. Examiner interprets cutting to be deleting.

Regarding claim 70, Applicant argues that Web Writer in combination with Web Writer II doesn't disclose position information regarding how components are placed on the template.

Examiner disagrees. Web Writer discloses, under the heading Web Writer editor, that *"…Web Writer inserts a new object of the requested class into the content tree data structure at the position designated by the insertion point and sets this new object to be the selected object…"*

Regarding claim 72, applicant argues that Web Writer doesn't disclose reloading edited pages. Examiner disagrees, Web Writer on page, right above the head, Discussion of the Web Writer Approach discloses, adding refresh meta tags to the pages, which examiner interprets would enable to refresh the edited content. Web Writer also discloses under the heading Constructing a Web Writer Application

"…two final steps of application-building: saving and loading a stack and running…"

Regarding claim 73, Applicant argues that there is no page generation in the web generator taught by Web Writer.

Examiner disagrees, Web Writer in his abstract teaches, "creating both Web pages and programs that generate Web pages…"

Regarding Applicant's arguments in claims 75-89, Applicant similarly rehashes arguments as previously addressed above.

Regarding Applicants argument for improper combination of both Web Writer I and Web Writer II in the 103 rejection in claim 77, Examiner disagrees. Both Web Writer articles are analogous art and both deal with Web editing and web page generation, and Web Writer II essentially offers an improvement over the initial Web Writer version and hence would considerably make that combination regarding implementing certain limitations in applicants claim not found in Web Writer.

Applicant argues in claim 80 that Web Writer doesn't disclose storing session information, again Examiner believes Web Writer does, under the Heading <u>Related Work , Web Writer discloses</u>, "saving and restoring the state of runtime data structures."

Applicant argues that in claim 81, that Web Writer doesn't disclose "instructions to decide about exclusions of components".

Again Examiner believes Web Writer does in fact disclose there limitations. Under the heading, <u>Web Writer Application model</u> it is disclosed that,

"…where each page can contain both static content that is always present and computed content that is generated on the fly…" as interpreted the computed content can be produced on the fly and isn't always present in comparison to the static content. And also regarding Applicants argument about nested components, the dynamic portions of the template are set up as place holders and hence examiner interprets this to be nested within the template since the dynamic portions is executed on the fly within the placeholder areas on the template. Please see Web Writer et seq.

Regarding claims 83-85, Applicant is rehashing arguments for previously addressed limitations above. Please see previous arguments pertaining to editor and components.

Regarding claim 88, it rehashes similar arguments as addressed above in claim 81 regarding exclusion.

Regarding claim 91, Applicant is arguing for an unclaimed merit of distinction.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., a second browsers) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Therefore Applicants argument is moot.

Regarding claims 92 – 96, Applicants arguments disclose previously addressed limitations above.

Regarding claim 115, passing information to the editor, see dynamic areas of editor as disclosed by Web Writer on page 2.

Regarding claims 116 – 119, Applicants arguments disclose previously addressed limitations above.

Regarding Applicant's arguments in claim 121, including extension for use by the editor, see Web Writer's disclosure on page 2, which shows under the heading related work, "Web Writer can show HTML tags mixed in with a preview of formatting, including the layout of tables, font changes and so on". Examiner interprets Applicants term of extension to mean the HTML tags as disclosed.

Regarding Applicant's arguments in claim 122 that We Writer doesn't disclose wherein the extension has to be part of the components in the editor form. Examiner disagrees, In Web Writer I, second page, Web Writer discloses, under the heading Related Work, that allowing extension tags is well known.

Regarding Applicants argument in claim 123, Examiner interprets the tag name to be the tag extensions and HTML tag identified in Web Writer I.

Regarding claim 124, Applicant rehashes arguments already addressed above.

Regarding argument in claims 126– 127, Applicant rehashes similar arguments with regards to running during editing which has already been addressed above in claim 1.

## **Pertinent Art**

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Fausitini USPN 5,842,020, also discloses dynamic editing.

## Correspondence Information

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chuck Kendall whose telephone number is 571-272-3698. The examiner can normally be reached between Monday and Thursday, at 11:00 am - 4:300pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Tuan Dam can be reached on

571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Chuck O Kendall/

Primary Examiner, Art Unit 2192